IS 6733
Deep Learning on Cloud Platforms

Lecture 3

Deep Learning Overview

Dr. Yuanxiong Guo

Department of Information Systems and Cyber Security

Acknowledgement

- Some slides are adapted from the following course:
  - CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University
- Thanks to Prof. Fei-Fei Li for sharing the slides

Deep Learning

- What is “learning”?
  - Improving performance through experience
  - Getting a computer to do well on a task without manually building in competence

- What is “deep”?
  - Successive layers of increasingly meaningful representations on the input data

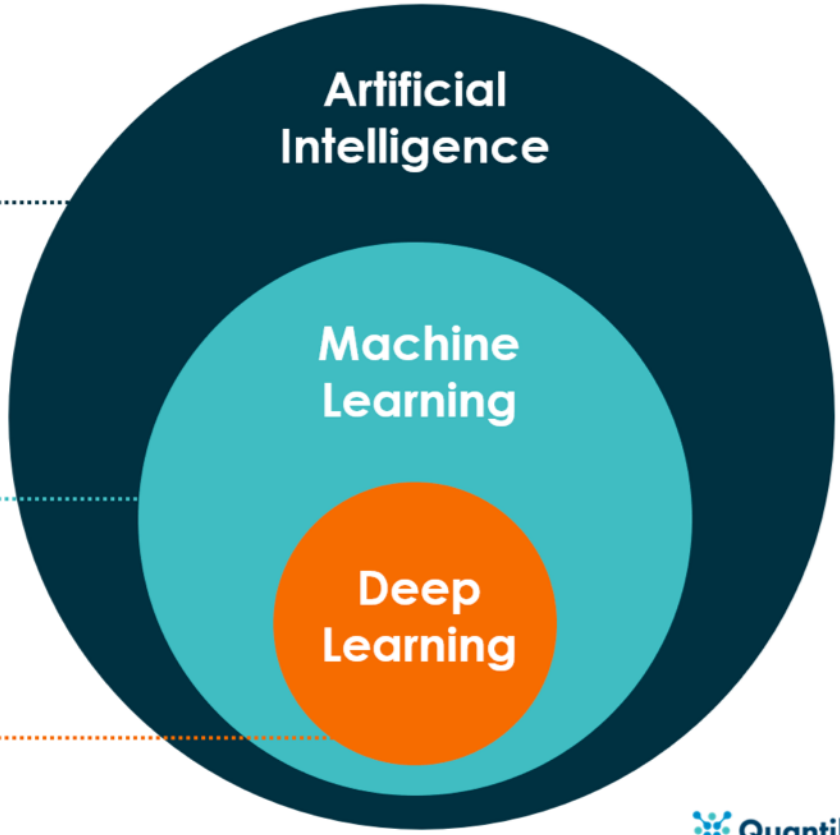
Artificial Intelligence, Machine Learning, and Deep Learning

Relationship between AI, ML, and DL

ARTIFICIAL INTELLIGENCE (AI)
Programming systems to perform tasks which usually require human intelligence.

MACHINE LEARNING (ML)
Training algorithms to solve tasks by pattern recognition instead of specifically programming them how to solve the task.

DEEP LEARNING (DL)
Training algorithms by using deep neural networks with multiple layers.



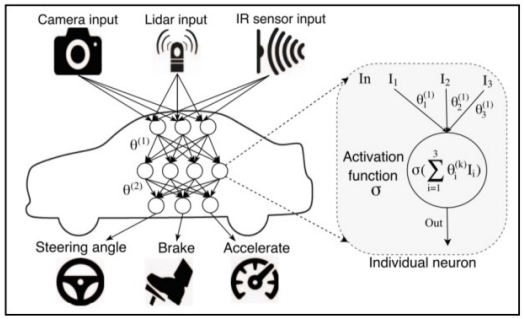
Quantib

Today AI is ubiquitous

- Automate routine labor
  - Search

- Understand speech
  - SIRI, Alexa

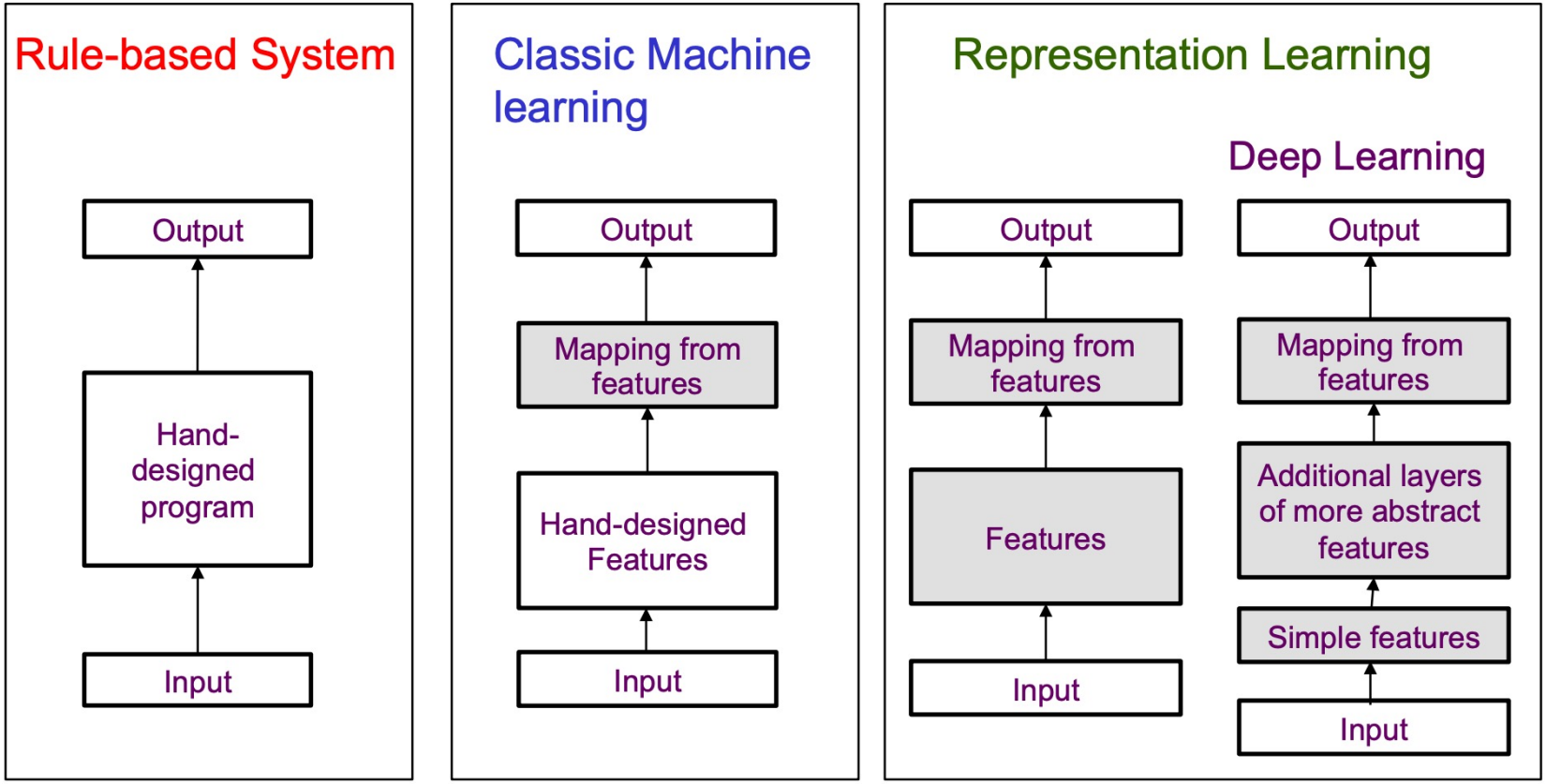
- Autonomous Vehicles



Challenge of AI

- Early successes of AI
  - Solved problems intellectually difficult for humans
    - Problems described by small set of rules
- True challenge of AI
  - Solve tasks easy for people but hard to describe formally
    - e.g., recognize spoken words, or faces in images
- Today's AI
  - It is about solving these more intuitive problems

Summary of AI Models



AI Paradigm Shift

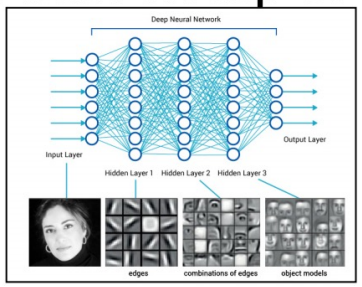
- Knowledge-based systems
  - Cannot perform simple recognition tasks
- Simple machine learning methods
  - Cannot perform complex recognition tasks
- Deep learning methods

Everyday life needs knowledge

- A person's everyday life requires immense amount of knowledge of the word
  - Much of this knowledge is intuitive and subjective
  - Difficult to articulate in a formal way
- Computers need to capture same knowledge to behave intelligently
- Key challenge of AI is how to get this informal knowledge into a computer

Solution to Intuitive Problems

1. Allow computers to learn from experience
   - By gathering knowledge from experience
     - Avoids need for human operators to specify knowledge that the computer needs
2. Understand the world as hierarchy of concepts
   - Thereby learn complicated concepts by building them out of simpler ones
     - A graph of how these concepts are built on top of each other is deep, with many layers



The Knowledge-based Approach

- Hard-coded knowledge in a formal language
  - Computer can reason about statements in these languages using inference rules
- Unwieldy process
  - Staff of human supervisors
  - People struggle formalize rules with enough complexity to describe the world

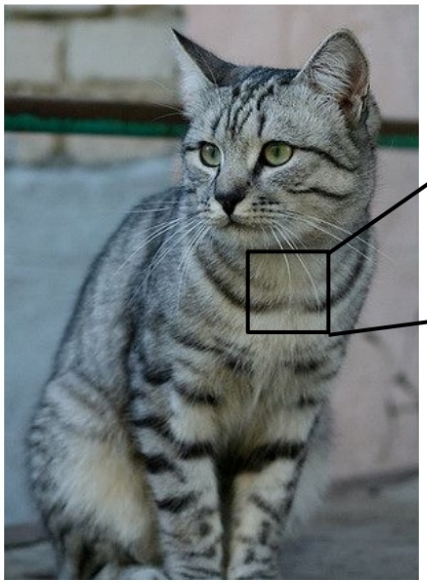
Example: Image Classification

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

→ cat

Example: Image Classification

The Problem: Semantic Gap




```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

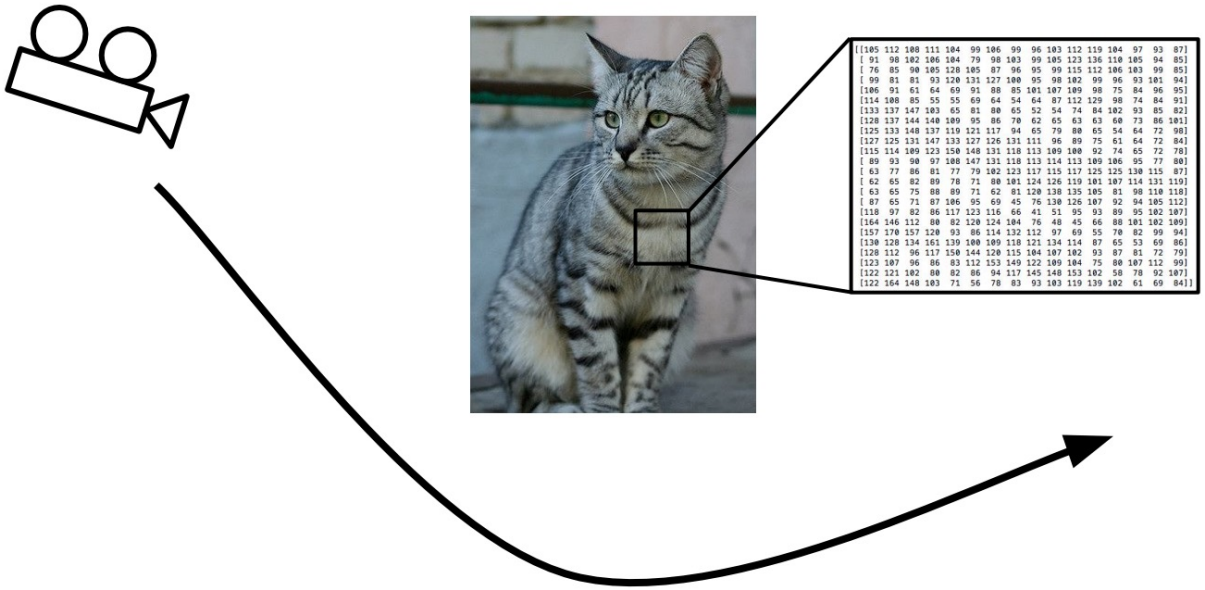
What the computer sees

An image is just a big grid of numbers between [0, 255]:

e.g. 800 x 600 x 3
(3 channels RGB)

Example: Image Classification

Challenges: Viewpoint variation



Example: Image Classification

Challenges: Background Clutter

This image is CC0 1.0 public domain

This image is CC0 1.0 public domain

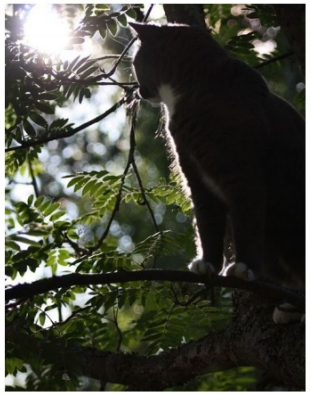
Example: Image Classification

Challenges: Illumination

This image is CC0 1.0 public domain

This image is CC0 1.0 public domain

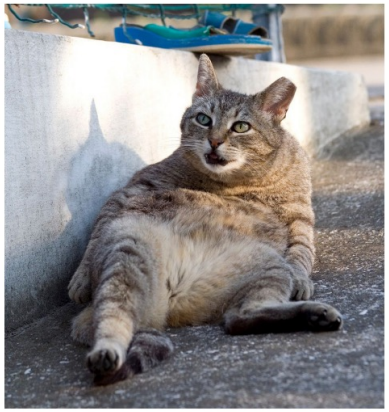
This image is CC0 1.0 public domain



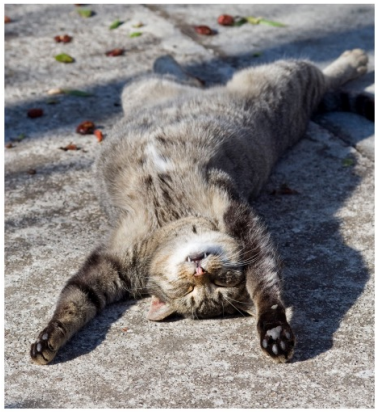
This image is CC0 1.0 public domain

Example: Image Classification

Challenges: Deformation

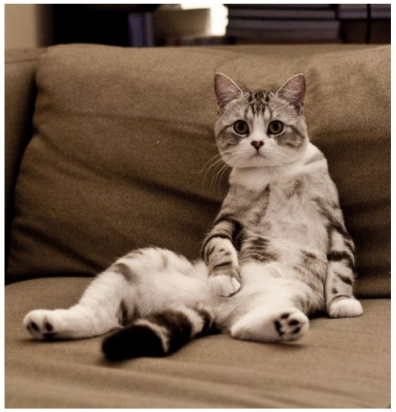


This image by Umberto Salvagnin is licensed under CC-BY 2.0



This image by Umberto Salvagnin is licensed under CC-BY 2.0

This image by sare bear is licensed under CC-BY 2.0



This image by Tom Thai is licensed under CC-BY 2.0

Example: Image Classification

Challenges: Occlusion

This image is CC0 1.0 public domain

This image is CC0 1.0 public domain

This image by jonsson is licensed under CC-BY 2.0

Example: Image Classification

Challenges: Intraclass variation

Example: Image Classification

An image classifier

```
def classify_image(image):
  # Some magic here?
  return class_label
```

Unlike e.g. sorting a list of numbers,

no obvious way to hard-code the algorithm for recognizing a cat, or other classes.

The ML Approach

- Difficulties of hand-coded approach suggests:
  - AI systems need ability to acquire knowledge by extracting patterns from raw data
- This approach is known as machine learning
  - It allowed tackling problems requiring knowledge of real world and make decisions that seem subjective
  - Simple ML algorithms
    - Logistic Regression
    - Naive Bayes

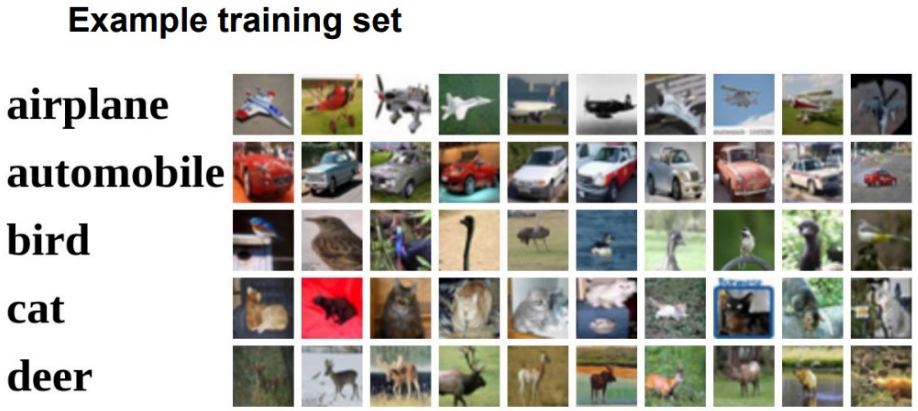
The ML Approach

Machine Learning: Data-Driven Approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

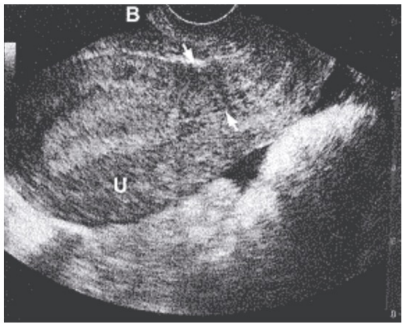
```
def train(images, labels):
  # Machine learning!
  return model
```

```
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```



Simple ML depends on representation

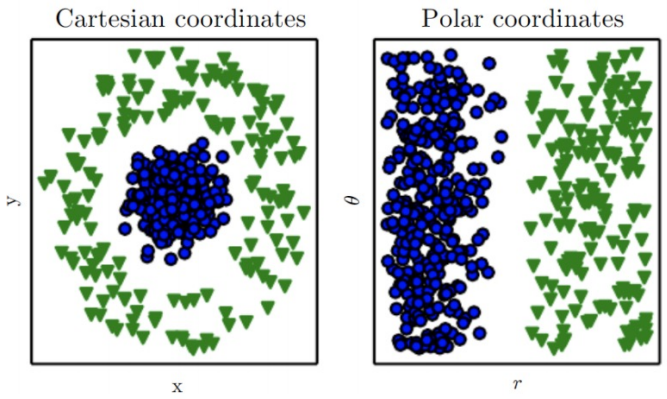
- Simple ML algorithms depend heavily on *representation* of given data
- Ex: Logistic regression to recommend delivery does not examine patient directly
  - Instead doctor provides information such presence of uterine scar (called a feature)
  - If logistic regression was provided with MRI scan, it could not make useful predictions



Individual pixels have negligible correlation with complications in delivery

Dependence on Representation

- This dependence appears in computer science and daily life
  - In CS: search is exponentially faster if data is structured and indexed intelligently
  - People: arithmetic on Arabic numerals 1,2,...9,10, easier than using Roman numerals I, II,…IX, X
  - Choice of representation has enormous effect on ML algorithm performance
  - Ex: straight line separation
    - Impossible in Cartesian
    - Simple straight line in polar



Designing right set of features

- Difficult to know what set of features are good for detecting a car in photographs
- Presence of wheel as a feature
  - Has a simple geometric shape but difficult to describe in terms of pixel values
  - Shadows, glare from metal parts, fender or object in front obscures



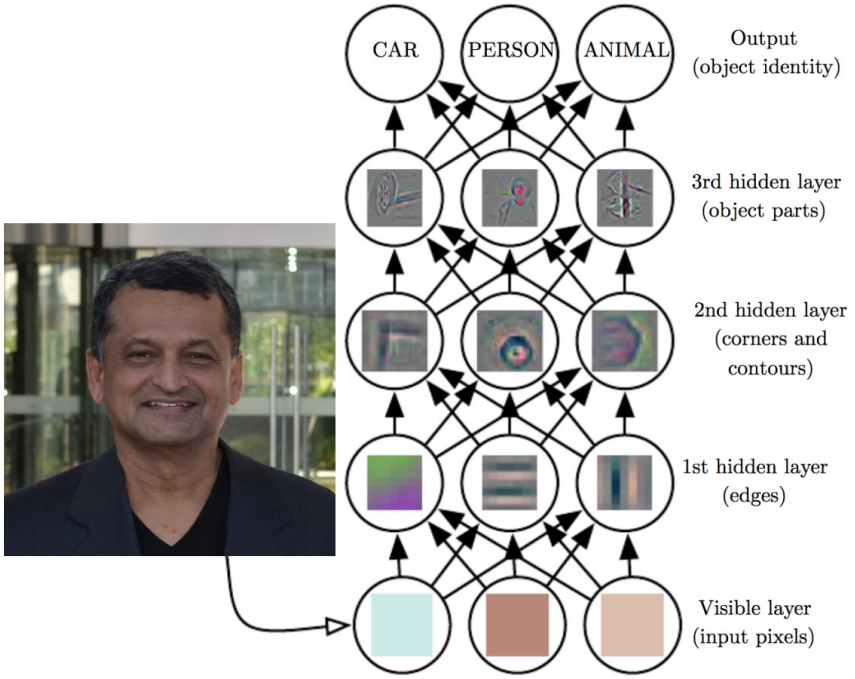
Representation Learning

- Solution: use ML to not only learn mapping from representation to output but representation itself
  - Better results than hard-coded representations

- Allows AI systems to rapidly adapt to new tasks
  - Designing features can take great human effort
  - Can take decades for a community of researchers

Feature Learning for Classification

- Function to map pixels to object identity is complicated
- Series of hidden layers extract increasingly abstract features
- Final decision made by a simple classifier



Characteristics of Deep Learning

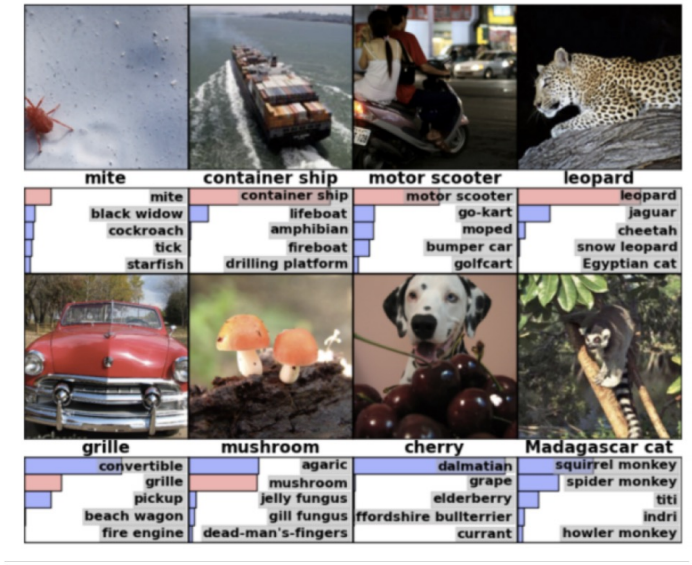
- A type of Machine Learning that improves with experience and data
- Only viable approach to building AI systems in real-world environments
- Obtains its power by a nested hierarchy of concepts
  - each concept defined by relationship to simpler concepts
  - More abstract representations computed in terms of less abstract ones

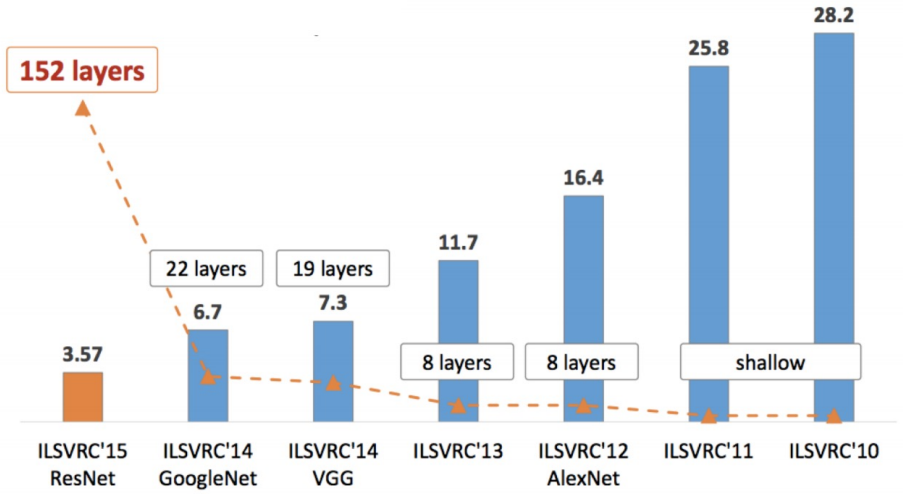
Success of Deep Learning

- Vision
- Speech and language
- Game
- Robotics
- Self-driving cars

Vision

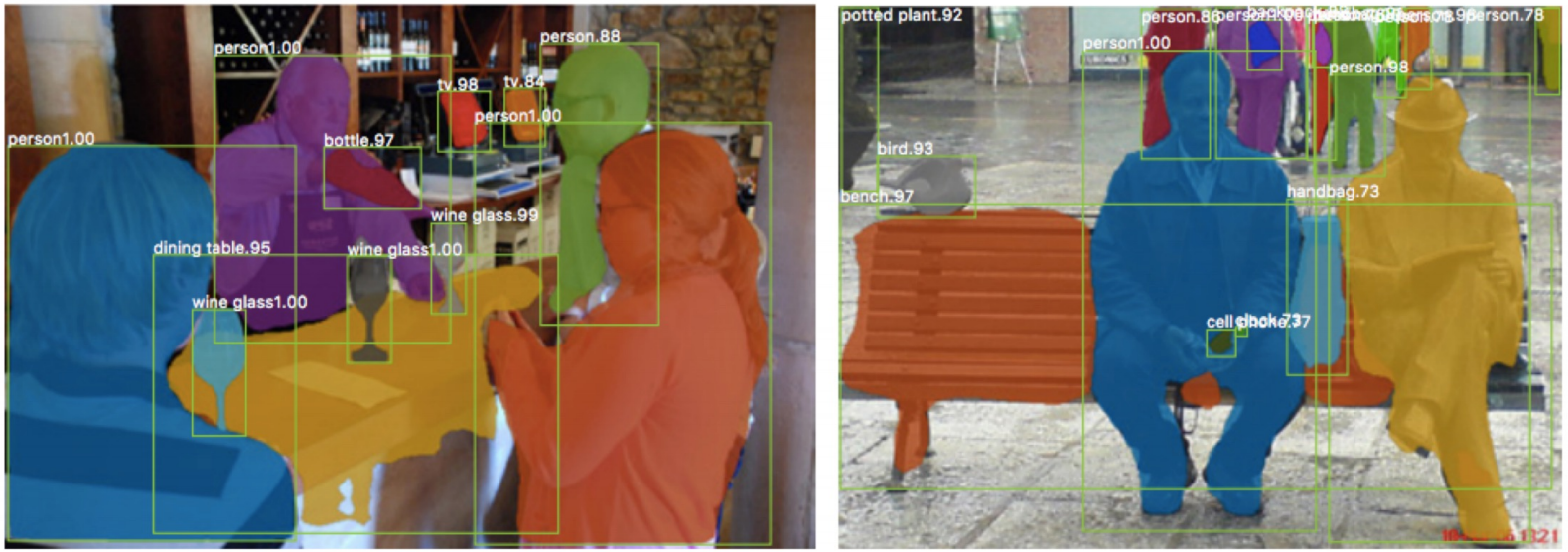
- ImageNet Challenge





Vision

- Object detection, instance segmentation

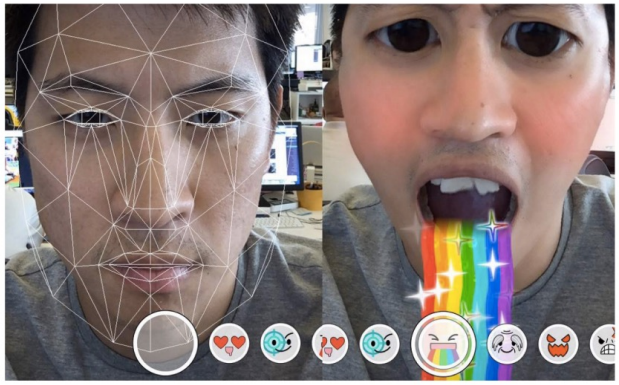


K. He, G. Gkioxari, P. Dollar, and R. Girshick, Mask R-CNN,
ICCV 2017 (Best Paper Award)

Vision

Facebook accessibility tools for the visually impaired

AI beats human pathologists at detecting cancer



Technology behind Snapchat lenses

Vision

Beijing bets on facial recognition in a big drive for total surveillance – Washington Post, 1/8/2018

Speech and natural language

Skype Translator

Break down the language barrier with your friends, family and colleagues.

Our online translator can help you communicate in 7 languages for voice calls, and in more than 50 languages while instant messaging.

Skype Translator uses machine learning. So the more you use it, the better it gets. Thanks for being patient as the technology graduates from Preview mode.

https://www.skype.com/en/features/skype-translator/

Google Translate App

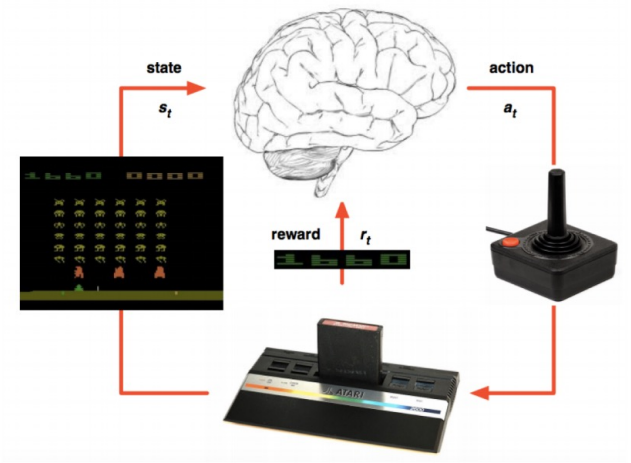
- Translate between 103 languages by typing
- Offline: Translate 52 languages when you have no Internet
- Instant camera translation: **Use your camera to translate text** instantly in 30 languages
- Camera Mode: Take pictures of text for higher-quality translations in 37 languages
- Conversation Mode: Two-way **instant speech translation** in 32 languages

https://play.google.com/store/apps/details?id=com.google.android.apps.translate&hl=en

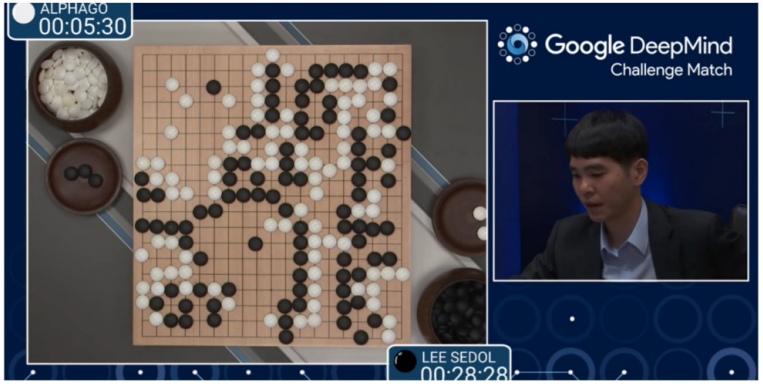
See also: The Great AI Awakening (New York Times Magazine)

Games

- 2013: DeepMind uses deep reinforcement learning to beat humans at some Atari games

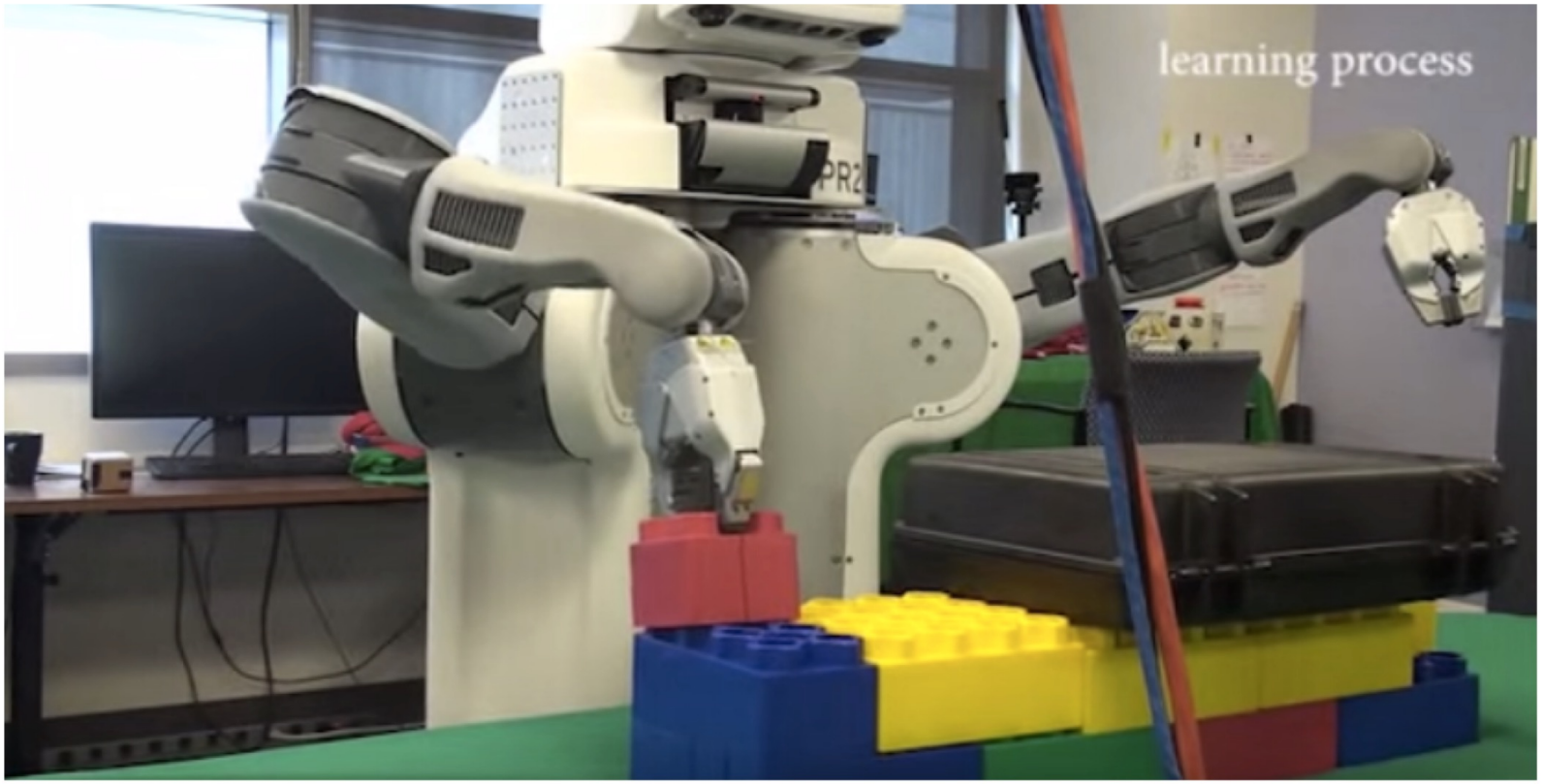


- 2016: DeepMind's AlphaGo system beats Go grandmaster Lee Sedol 4-1
- 2017: AlphaZero learns to play Go and chess from scratch



Robots

- End-to-end training of deep visuomotor policies



https://www.youtube.com/watch?v=Q4bMcUk6pcw

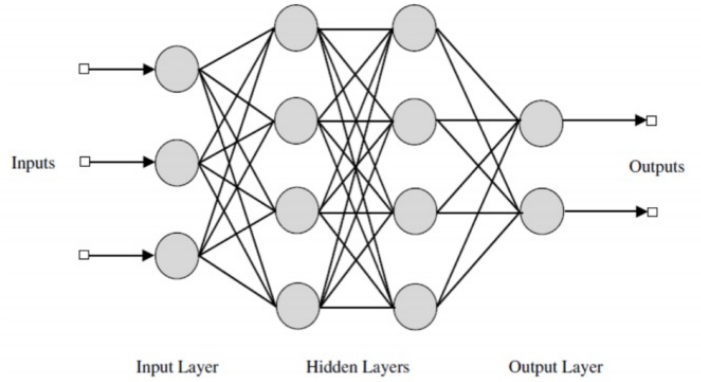
Self-driving cars

Image source

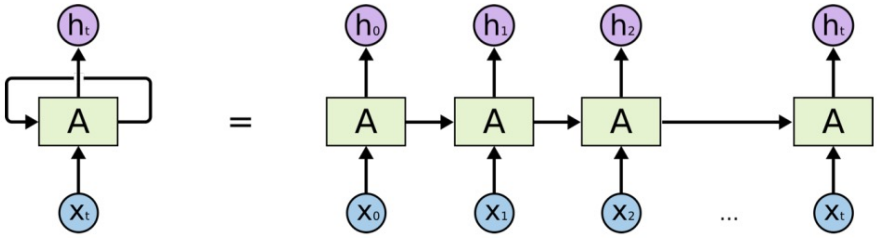
- Deep learning crucial for the global success of automotive autonomy – Automotive World, 6/26/2018

Basic Deep Learning Models

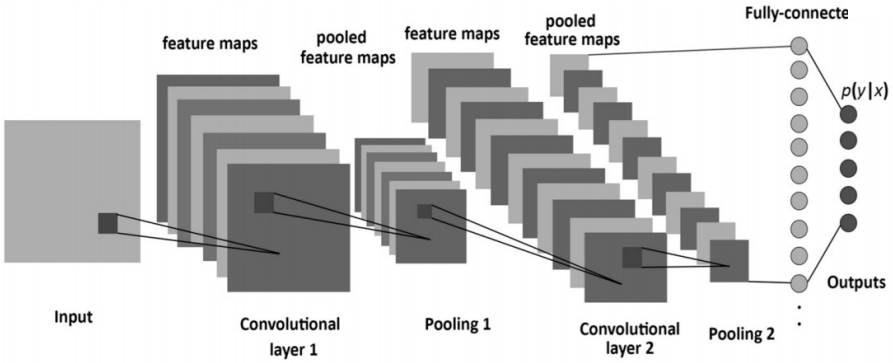
Multilayer neural networks, backpropagation



Recurrent networks and applications



Convolutional networks and applications



Before DL: A Brief History of ML

A Few Facts about DL

- DL has reached a level of public attention and industry investment never seen before in the history of AI

But,

- DL isn't the first successful form of ML
- Most of the ML algorithms used in the industry today aren't DL algorithms.
- DL isn't always the right tool for the job
  - Not enough data for DL to be applicable
  - Problems better solved by a different ML algorithm

Probabilistic Modeling

- The application of the principles of statistics to data analysis
- One of the earliest forms of ML and still widely used to this day
  - Naïve Bayes
    - Classifiers based on applying Bayes' theorem while assuming the features in the input data are all independent (a strong, or "naïve" assumption)
  - Logistic Regression
    - Often the first thing a data scientist will try on a dataset to get a feel for the classification task at hand

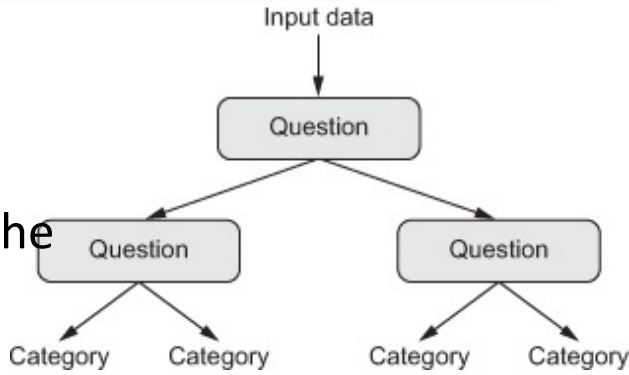
Early Neural Networks

- The core ideas of neural networks were investigated in toy forms as early as 1950s
- The major difficulty at that time was an efficient way to train large neural networks
- Changed in mid-1980s due to the invent of *backpropagation* algorithm
- The first successful application of neural nets came in 1980 from Bell Labs
  - Yann LeCun applied CNN and backpropagation to classify handwritten digits
  - The resulting network (LeNet) was used by USPS in the 1990s to automate the reading of ZIP code on mail envelops

Kernel Methods

- As neural networks started to gain some attention in the 1990s, *Kernel methods* rose to fame and quickly sent neural nets back to oblivion.
- *Kernel methods:* a group of classification algorithms where
  1. Data is mapped to a new high-dimensional representation where the decision boundary can be expressed as a hyperplane (using kernel function to efficient compute the distance between pairs of points in the new representation)
  2. A good decision boundary is computed by trying to maximize the distance between the hyperplane and the closest data points from each class
- SVMs exhibited state-of-the-art performance on simple classification problems with rigorous theory support
  - Did not provide good results for perceptual problems such as image classification (shallow method)
  - Requires feature engineering before being applied

Decision Trees, Random Forests, and Gradient Boost Machines



- *Decision trees* are flowchart-like structures
  - Easy to visualize and interpret
  - Began to receive significant research interest in the 2000s, and by 2010 often preferred to kernel methods
- *Random forest* involves building a large number of specialized decision trees and then ensembling their outputs
  - Applicable to a wide range of problems
  - Almost always the second-best algorithm for any shallow ML task
- *Gradient boosting machines:* similar to a random forest but uses gradient boosting
  - A way to improve any ML model by iteratively training new models that specialize in addressing the weak points of the previous models
  - One of the best algorithm for dealing with nonperpetual data today

Back to Neural Networks

- Around 2010, neural networks were almost completely shunned by the scientific community at large
- The watershed moment came in 2012: Geoffrey Hinton at UToronto achieved 83.6% vs. 74.3% in ImageNet challenge using deep convolutional neural networks.
- Since 2012, deep convolutional neural networks have become the go-to algorithm for all computer vision tasks, or more generally, all perceptual tasks.

The Modern Machine Learning Landscape

- Two dominant approaches used by practitioners: *gradient boosting machines* and *deep learning*
  - *Gradient boosting* is used for problems where structured data is available (shallow-learning problems)
    - XGBoost library (Python and R)
  - *Deep learning* is used for perceptual problems such as image classification
    - Keras library (Python and R)

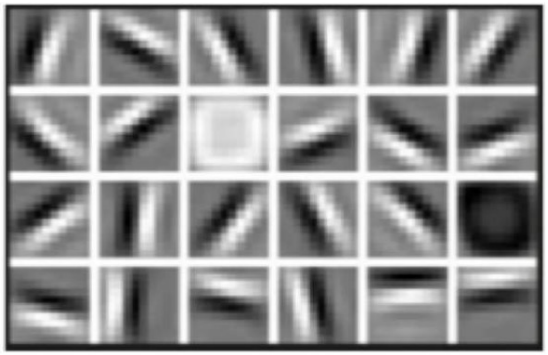
Why Deep Learning? Why Now?

Why Deep Learning?

Hand engineered features are time consuming, brittle and not scalable in practice

Can we learn the **underlying features** directly from data?

Low Level Features



Lines & Edges

Mid Level Features

Eyes & Nose & Ears

High Level Features

Facial Structure

Why Now?

1952 Stochastic Gradient Descent

1958 Perceptron
- Learnable Weights

1986 Backpropagation
- Multi-Layer Perceptron

1995 Deep Convolutional NN
- Digit Recognition

Neural Networks date back decades, so why the resurgence?

1. Big Data
- Larger Datasets
- Easier Collection & Storage

IMAGENET

WIKIPEDIA
The Free Encyclopedia

2. Hardware
- Graphics Processing Units (GPUs)
- Massively Parallelizable

3. Software
- Improved Techniques
- New Models
- Toolboxes

TensorFlow

Deep Learning Hardware

Inside a computer

Spot the CPU!

(central processing unit)

Spot the GPUs!

(graphics processing unit)

This image is in the public domain

NVIDIA vs AMD

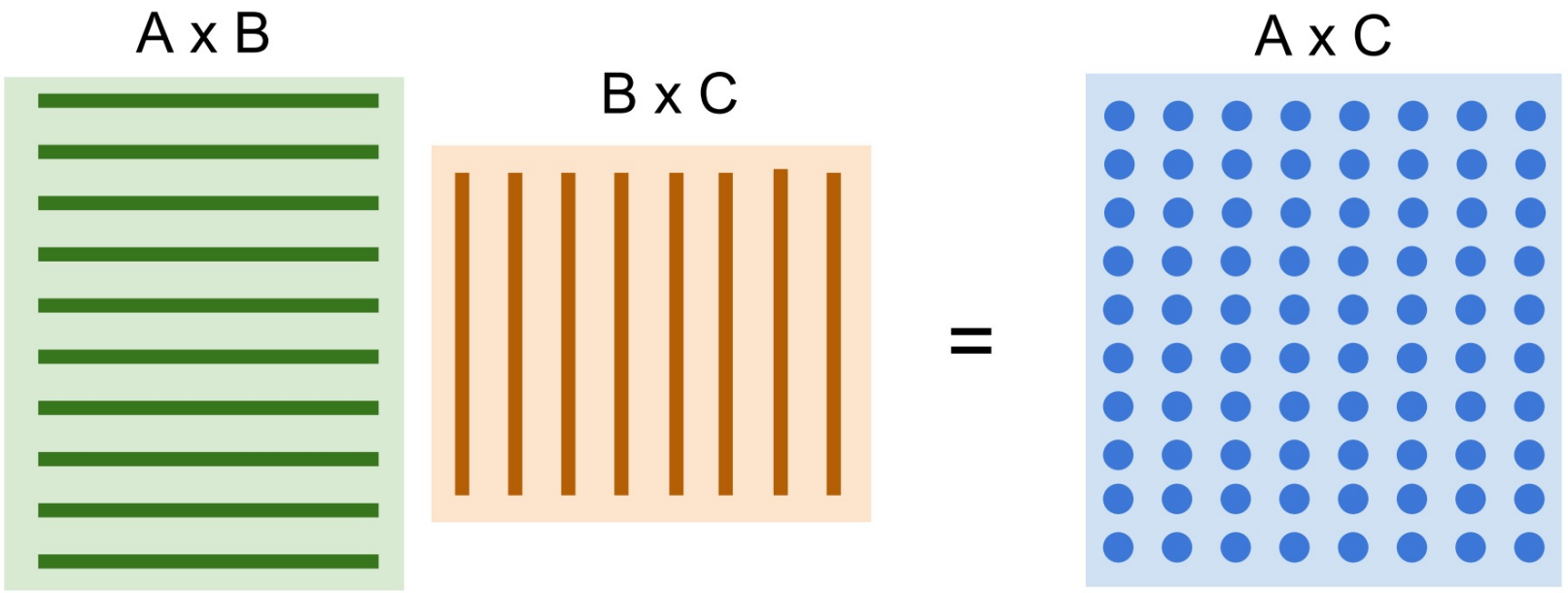
CPU vs GPU

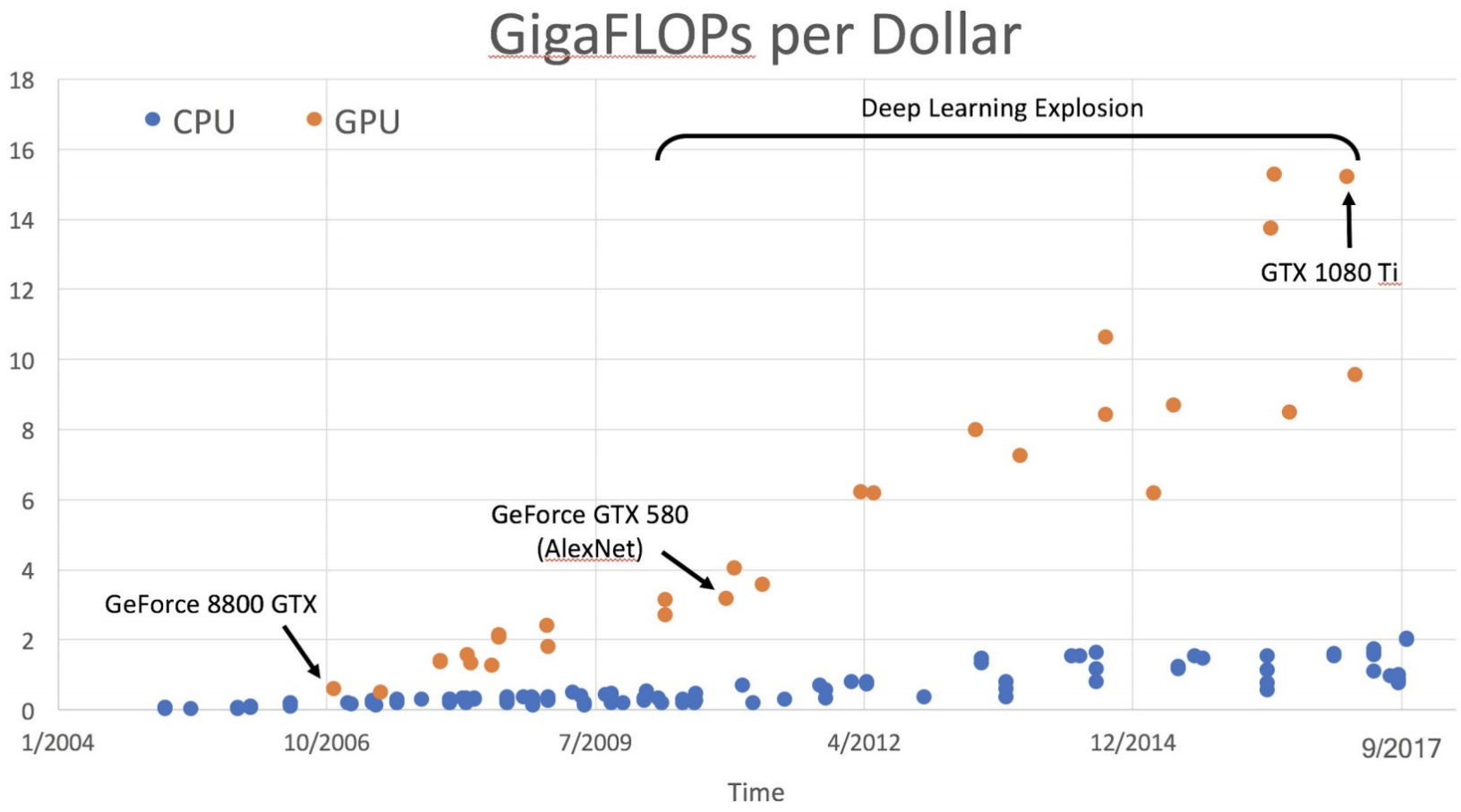
	Cores	Clock Speed	Memory	Price	Speed
CPU (Intel Core i7-7700k)	4 (8 threads with hyperthreading)	4.2 GHz	System RAM	$385	~540 GFLOPs FP32
GPU (NVIDIA RTX 2080 Ti)	3584	1.6 GHz	11 GB GDDR6	$1199	~13.4 TFLOPs FP32

CPU: Fewer cores, but each core is much faster and much more capable; great at sequential tasks

GPU: More cores, but each core is much slower and "dumber"; great for parallel tasks

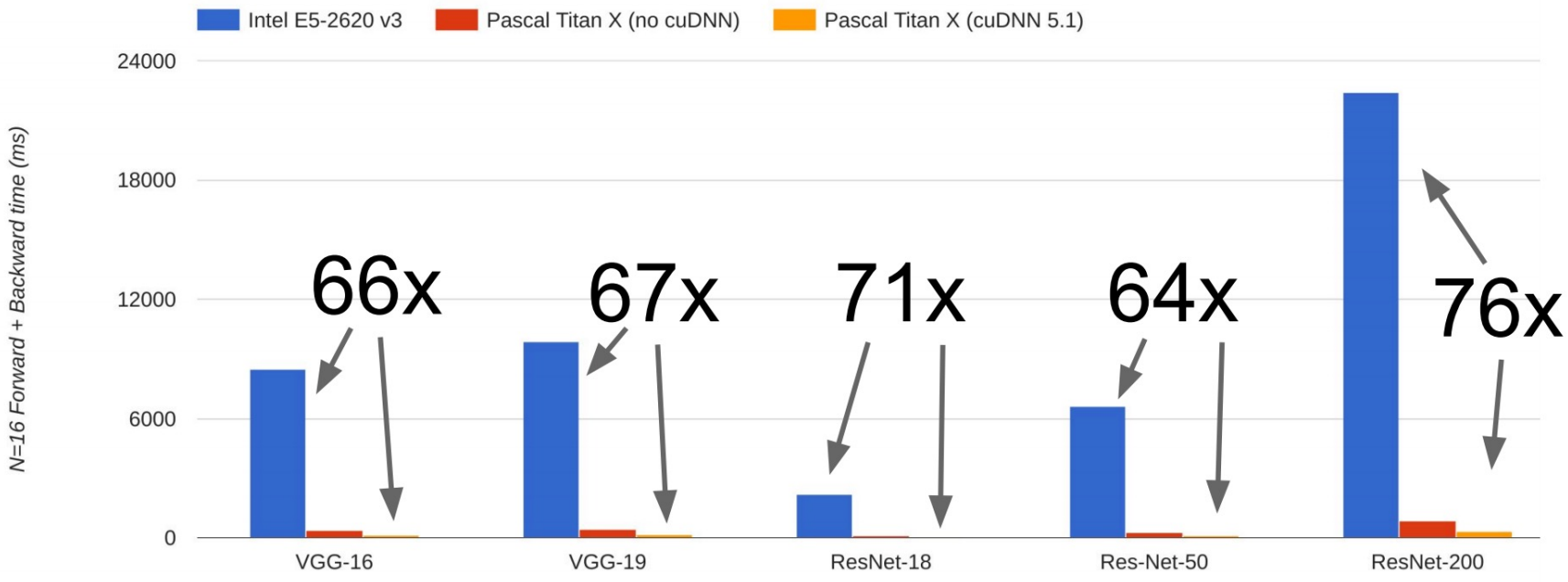
Example: Matrix Multiplication




GigaFLOPs per Dollar
CPU
GPU
Deep Learning Explosion
GTX 1080 Ti
GeForce GTX 580
(AlexNet)
GeForce 8800 GTX
18
16
14
12
10
8
6
4
2
0
1/2004
10/2006
7/2009
4/2012
12/2014
9/2017
Time

CPU vs GPU in practice

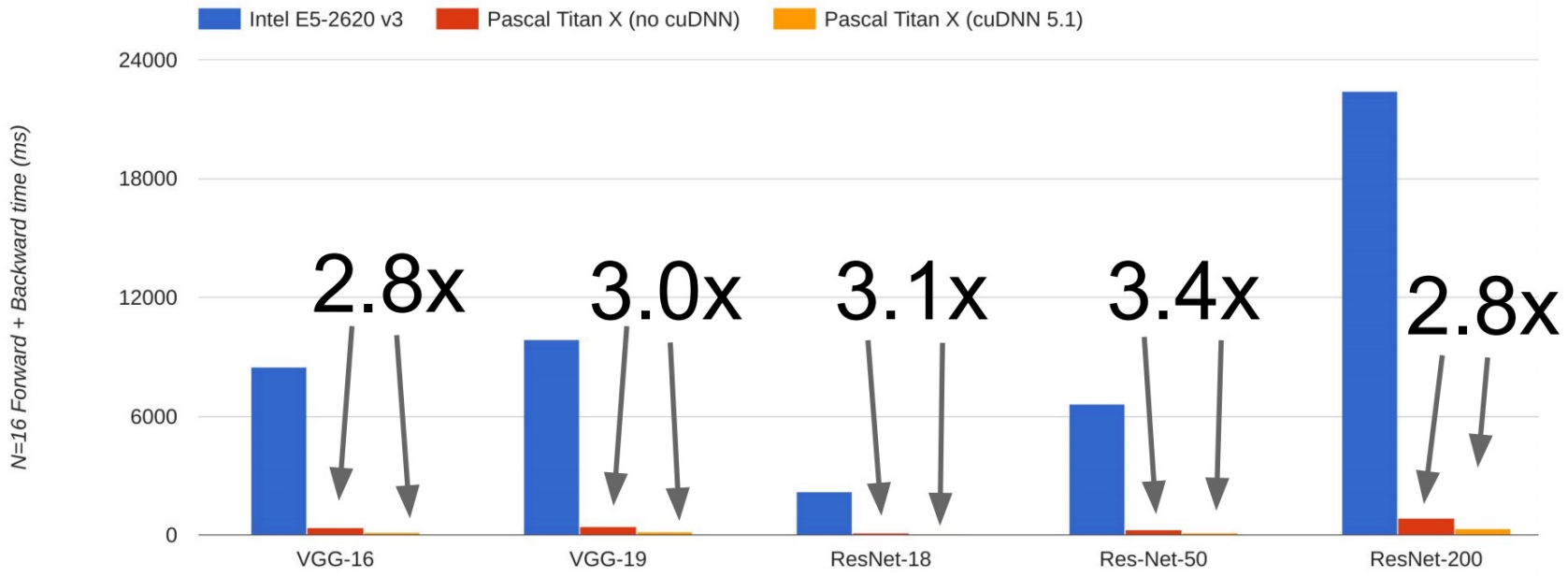
(CPU performance not well-optimized, a little unfair)



Data from https://github.com/jcjohnson/cnn-benchmarks

CPU vs GPU in practice

cuDNN much faster than
“unoptimized” CUDA



Data from https://github.com/jcjohnson/cnn-benchmarks

CPU vs GPU

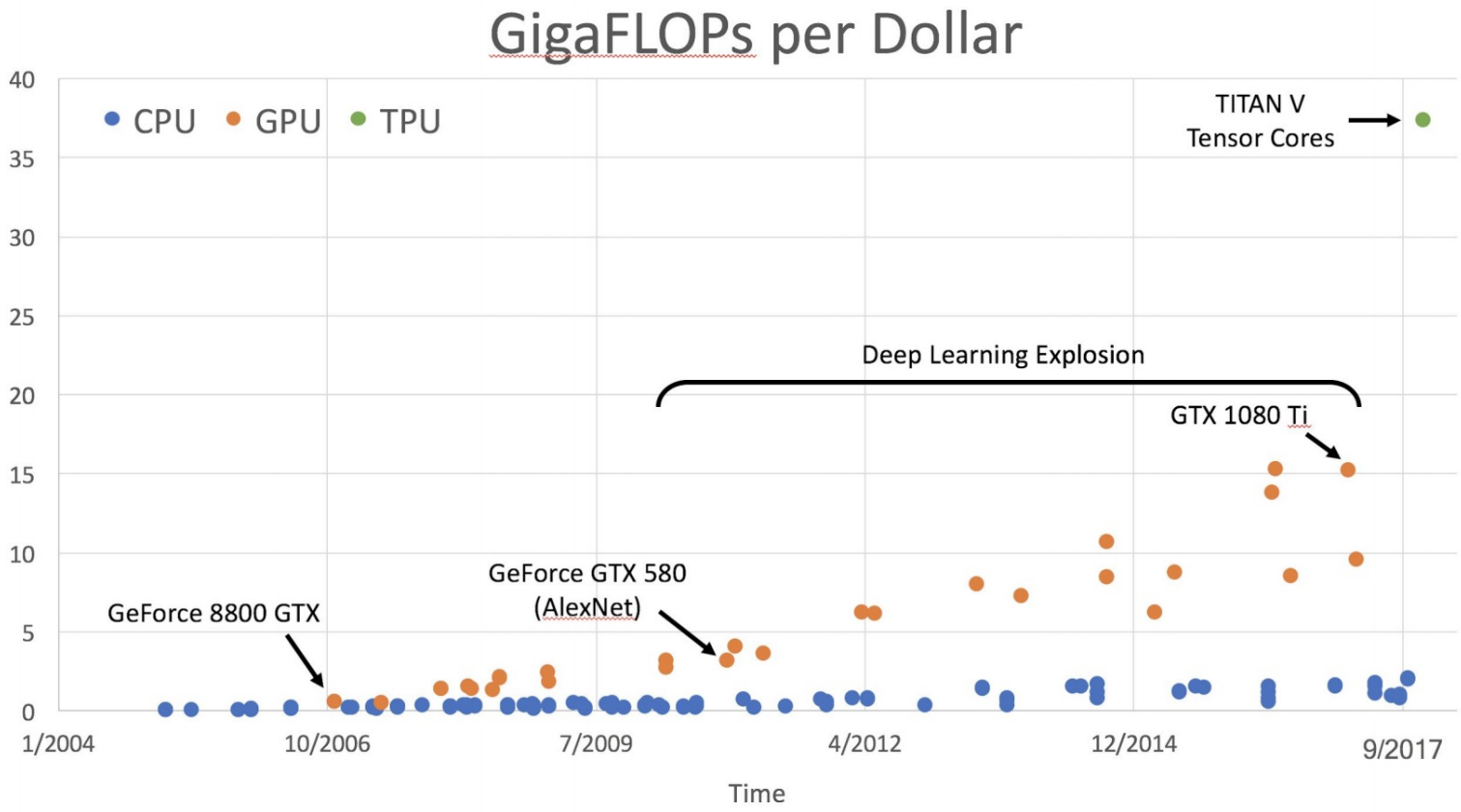
	Cores	Clock Speed	Memory	Price	Speed
CPU (Intel Core i7-7700k)	4 (8 threads with hyperthreading)	4.2 GHz	System RAM	$385	~540 GFLOPs FP32
GPU (NVIDIA RTX 2080 Ti)	3584	1.6 GHz	11 GB GDDR6	$1199	~13.4 TFLOPs FP32
TPU NVIDIA TITAN V	5120 CUDA, 640 Tensor	1.5 GHz	12GB HBM2	$2999	~14 TFLOPs FP32 ~112 TFLOP FP16
TPU Google Cloud TPU	?	?	64 GB HBM	$4.50 per hour	~180 TFLOP

NOTE: TITAN V isn't technically a "TPU" since that's a Google term, but both have hardware specialized for deep learning

CPU: Fewer cores, but each core is much faster and much more capable; great at sequential tasks

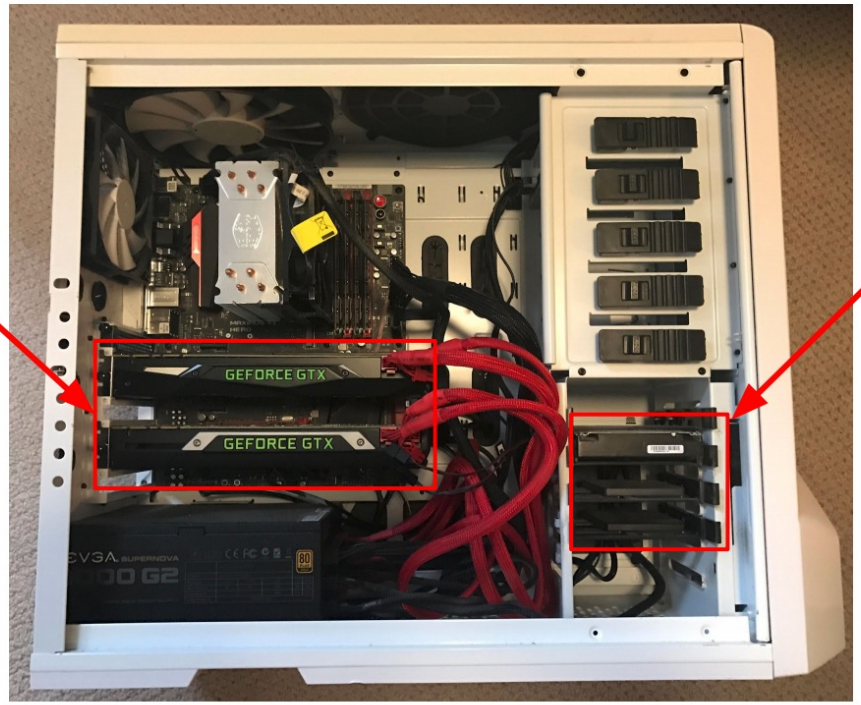
GPU: More cores, but each core is much slower and "dumber"; great for parallel tasks

TPU: Specialized hardware for deep learning


GigaFLOPs per Dollar
CPU
GPU
TPU
TITAN V
Tensor Cores
Deep Learning Explosion
GTX 1080 Ti
GeForce GTX 580
(AlexNet)
GeForce 8800 GTX
40
35
30
25
20
15
10
5
0
1/2004
10/2006
7/2009
4/2012
12/2014
9/2017
Time

CPU / GPU Communication

Model is here



Data is here

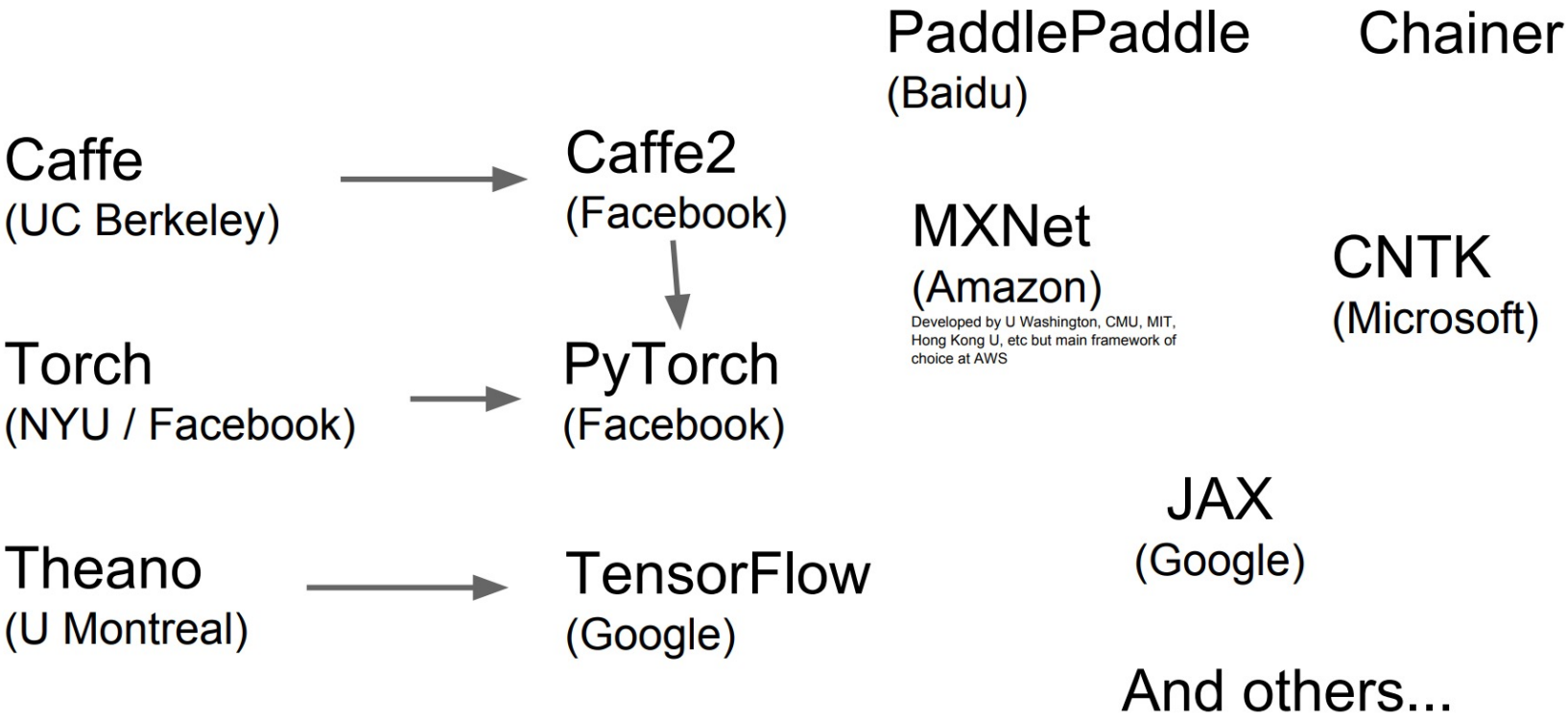
If you aren't careful, training can bottleneck on reading data and transferring to GPU!

Solutions:

- Read all data into RAM
- Use SSD instead of HDD
- Use multiple CPU threads to prefetch data

Deep Learning Software

A zoo of frameworks!

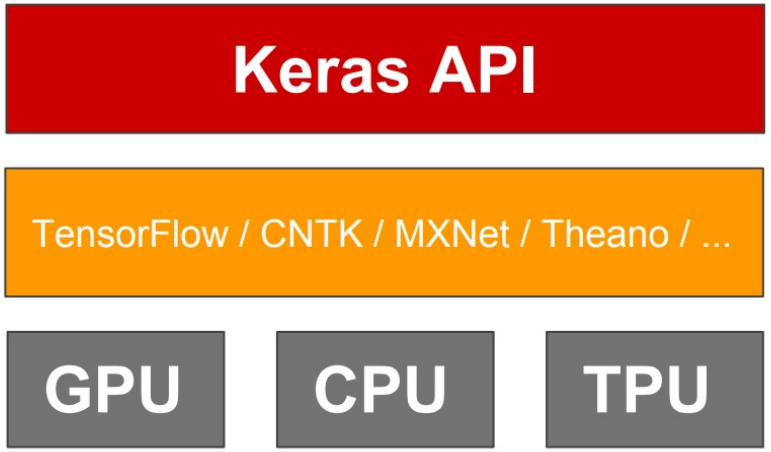


The point of deep learning framework

- Quick to develop and test new ideas
- Automatically compute gradients
- Run it all efficiently on GPU (wrap cuDNN, cuBLAS, etc)

Keras is multi-backend, multi-platform

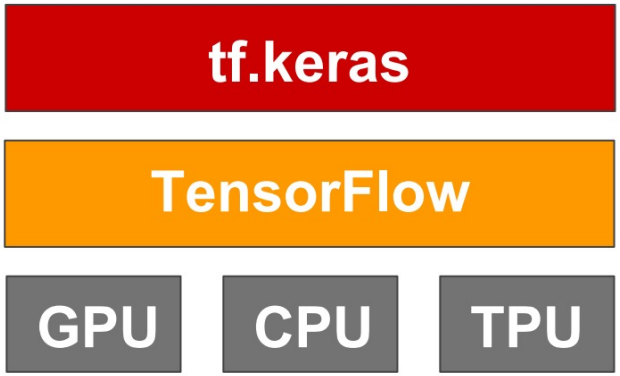
- Develop in Python, R
  - On Unix, Windows, OSX
- Run the same code with…
  - TensorFlow
  - CNTK
  - Theano
  - MXNet
  - PlaidML
  - - ??
- CPU, NVIDIA GPU, AMD GPU, TPU...



Our Focus – Keras API of TensorFlow

- Keras is the official high-level API of TensorFlow

  - tensorflow.keras (tf.keras) module
  - Part of core TensorFlow since v1.4
  - Full Keras API
  - Better optimized for TF
  - Better integration with TF-specific features
    - Estimator API
    - Eager execution
    - etc.




```
# import tf.keras as part of your TensorFlow program setup:
import tensorflow as tf
from tensorflow import keras
```

Reading Materials

- Chapter 1 of [T2]